

IB113 Úvod do programování a algoritmizace

Cvičení 2

Jaromír Plhák



Opakování

2.3.3. Pyramida

```
def pyramid(n):  
    pass  
  
"""  
    #  
    # # #  
    # # # # #  
    # # # # # # #  
    # # # # # # # # #  
    """  
pyramid(5)
```

2.1.4 Fibonacci

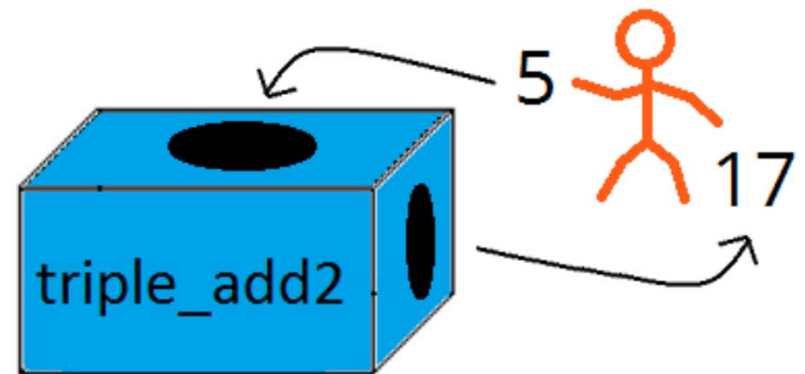
```
def fibonacci(n):  
    pass  
  
# 1 1 2 3 5 8 13 21 34 55  
fibonacci(10)
```

Funkce

- Krabíčka, která vezme vstup, něco s ním udělá a vrátí výstup

```
def triple_add2 (x):  
    return 3 * x + 2
```

```
>>> triple_add2 (5)  
17
```



Návratová hodnota

- Funkce **vrátí** výstup, tzv. návratovou hodnotu, volajícímu
- Všimněte si, že příkaz `return` hodnota okamžitě končí běh funkce a vrací hodnotu volající funkci

```
def triple_add2 (x):  
    return 3 * x + 2
```

```
>>> return_value = triple_add2 (5)  
>>> return_value  
17
```

print vs. return

```
def triple_add2_return (x):  
    return 3*x + 2
```

```
def triple_add2_print (x):  
    print (3* x + 2)
```

```
>>> return_value = triple_add2_return (5)
```

```
>>> return_value
```

```
17
```

```
>>> return_value = triple_add2_print (5)
```

```
17
```

```
>>> return_value
```

```
None
```

Procedury a funkce – definice (opakování)

- Klíčové slovo def
- Mezera
- Jméno funkce
- Seznam parametrů v závorkách
 - Žádné parametry - prázdné závorky
- Dvojtečka
- Nový řádek
- Odsazení příkazu funkce

```
def my_funct (par1 , par2 , ...) : # header
    statement_1 # body
    ...
    statement_n
```

Příklad funkce

```
def greater_less_equal_5 ( number ):  
    if number > 5:  
        return 1  
    elif number < 5:  
        return -1  
    else :  
        return 0
```

```
>>> a = greater_less_equal_5 (4)  
>>> b = greater_less_equal_5 (5)  
>>> c = greater_less_equal_5 (6)
```

Lokální platnost proměnných

```
def double (n):  
    return 2 * n  
def triple (n): # Different n than before  
    return 3 * n  
def foo (a, b):  
    return double (a) + triple (b)
```

```
>>> foo (5, 7)  
>>> double (5) + triple (7)  
>>> 10 + triple (7)  
>>> 10 + 21  
>>> 31
```


Funkce knihovny math

- Je nutné načítání modulu: `from math import *`
 - Pozor na hvězdičkové importy
- Zaokrouhlování: `round()`, `ceil()`, `floor()`
- Goniometrie: `sin()`, `cos()`, . . .
- Mocniny a logaritmy: `exp()`, `log()`, `sqrt()`, . . .
- Konstanty: `pi`, `e`
- <https://docs.python.org/3.5/library/math.html>

Ladění (debugging)

- Přečíst a pochopit chybovou hlášku
- Mentální trasování kódu
- Papír a tužka (!)
- Ladicí výpisy
 - Pomocné výpisy, které nám říkají, v jakém stavu se program nachází (např. v každé iteraci cyklu vypisujeme obsah proměnných)
- Debugger
 - Nástroj, pomocí něhož je možné postupně provádět jednotlivé kroky
 - Programu, sledovat hodnoty proměnných apod.
 - Typicky součástí vývojového prostředí

Debugger v PyCharmu

- Nastavení řádků, kde se má výpočet zastavit: breakpoints
 - Stačí kliknout vpravo vedle čísla řádku
- Spuštění programu s debuggerem
 - Místo Run použít Debug
- Hodnoty proměnných v okně Variables
 - Přidání nového výrazu, tlačítka + a –
- Krokování
 - Step Over – krok na další řádek
 - Step Into – vstoupí dovnitř volání funkce
 - Step Out – vynoří se z volání funkce
 - Resume Program – běží až po další breakpoint

Úkoly pro dnešek

- 3.2.2. Počet dělitelů
 - Napište funkci `divisors_count(n)`, která vrátí počet dělitelů čísla `n`
- 3.2.3. Je prvočíslo
 - Napište funkci `is_prime(n)`, která vrátí `True` pokud je číslo `n` prvočíslo, jinak `False`
- 3.2.6. Prvních `k` prvočísel
 - Napište funkci `primes(count)`, která vypíše prvních `count` prvočísel
 - Využijte návratovou hodnotu funkce pro test prvočíselnosti
- 3.2.7. Prvních `k` prvočíselných dvojčat¶
 - Napište funkci `twin_primes(count)`, která vypíše prvních `count` prvočíselných dvojčat

Zadání domácí úlohy

- V ISu v odevzdávárně
 - https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006396/72006443/hw01_zadani.py
- Veškeré informace v souboru
- Odevzdejte pouze tento soubor do odevzdávárny
 - <https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006396/72006429/> (skupina 1)
 - <https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006400/72006450/> (skupina 2)
- Do 13. 10. 2017, 23:59
 - Výjimečně do pátku, kvůli využití řetězců